

AD-A218 006

DTIC FILE COPY

4

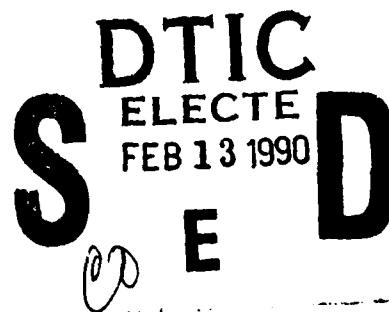
RADC-TR-89-259, Vol VIII (of twelve)
Interim Report
October 1989



NORTHEAST ARTIFICIAL INTELLIGENCE CONSORTIUM ANNUAL REPORT - 1988 Artificial Intelligence Applications to Speech Recognition

Syracuse University

Harvey E. Rhody, Thomas R. Ridley, John A. Biles



APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

This effort was funded partially by the Laboratory Director's fund.

**ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700**

00 02 12 190

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Services (NTIS) At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-89-259, Vol VIII (of twelve) has been reviewed and is approved for publication.

APPROVED:

John G. Parker

JOHN G. PARKER
Project Engineer

APPROVED:

Walter J. Senus

WALTER J. SENUS
Technical Director
Directorate of Intelligence & Reconnaissance

FOR THE COMMANDER:

Igor G. Plonisch

IGOR G. PLONISCH
Directorate of Plans & Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRAA) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS N/A		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) N/A			5. MONITORING ORGANIZATION REPORT NUMBER(S) RADC-TR-89-259, Vol VIII (of twelve)		
6a. NAME OF PERFORMING ORGANIZATION Northeast Artificial Intelligence Consortium (NAIC)		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Rome Air Development Center (COES)		
6c. ADDRESS (City, State, and ZIP Code) Science & Technology Center, Rm 2-296 111 College Place, Syracuse University Syracuse NY 13244-4100			7b. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Rome Air Development Center		8b. OFFICE SYMBOL (If applicable) COES	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F30602-85-C-0008		
8c. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. 62702F	PROJECT NO. 5581	TASK NO. 27
			WORK UNIT ACCESSION NO. 13		
11. TITLE (Include Security Classification) NORTHEAST ARTIFICIAL INTELLIGENCE CONSORTIUM ANNUAL REPORT - 1988 Artificial Intelligence Applications to Speech Recognition					
12. PERSONAL AUTHOR(S) Harvey E. Rhody, Thomas R. Ridley, John A. Biles					
13a. TYPE OF REPORT Interim		13b. TIME COVERED FROM Jan 88 to Dec 88		14. DATE OF REPORT (Year, Month, Day) October 1989	
15. PAGE COUNT 44					
16. SUPPLEMENTARY NOTATION This effort was funded partially by the Laboratory Directors' Fund. This effort was performed as a subcontract by Rochester Institute of Technology to Syracuse University, Office of Sponsored Programs.					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
23	02		Artificial Intelligence Speech Recognition		
05	07		Expert Systems Signal Processing		
			Phoneme Classification Knowledge-Based Systems		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) The Northeast Artificial Intelligence Consortium (NAIC) was created by the Air Force Systems Command, Rome Air Development Center, and the Office of Scientific Research. Its purpose is to conduct pertinent research in artificial intelligence and to perform activities ancillary to this research. This report describes progress that has been made in the fourth year of the existence of the NAIC on the technical research tasks undertaken at the member universities. The topics covered in general are: versatile expert system for equipment maintenance, distributed AI for communications system control, automatic photointerpretation, time-oriented problem solving, speech understanding systems, knowledge base maintenance, hardware architectures for very large systems, knowledge-based reasoning and planning, and a knowledge acquisition, assistance, and explanation system. The specific topic for this volume is the design and implementation of a knowledge-based system to read speech spectrograms.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL John G. Parker			22b. TELEPHONE (Include Area Code) (315) 330-4024		22c. OFFICE SYMBOL RADC (IRAA)

UNCLASSIFIED

Item 10. SOURCE OF FUNDING NUMBERS (Continued)

Program Element Number	Project Number	Task Number	Work Unit Number
62702F	5581	27	23
61102F	2304	J5	01
61102F	2304	J5	15
33126F	2155	02	10
61101F	LDFP	27	01

UNCLASSIFIED

Northeast Artificial Intelligence Consortium

1988 Annual Report

Volume 8

Artificial Intelligence Applications to Speech Recognition

Harvey E. Rhody
Thomas R. Ridley
John A. Biles

Rochester Institute of Technology
75 Highpower Road
Rochester, New York 14623



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Table Of Contents

8.1 Executive Summary.....	2
8.2 System Architecture	3
8.2.1 Software Architecture.....	3
8.2.2 Hardware Architecture.....	5
8.2.3 The ESPRIT System.....	5
8.3 Broad Phonetic Classification	9
8.3.1 Overview.....	9
8.3.2 Training.....	10
8.3.3 Cluster Analysis.....	10
8.3.4 Results of Classifying Unknown Data Samples.....	12
8.3.4.1 K-means and Maximum Likelihood Results.....	12
8.3.5 Neural Networks for Broad Phonetic Classification.....	13
8.4 Phonetic String Parsing.....	13
8.4.1 Background	13
8.4.2 Approaches To String Comparison.....	14
8.4.3 String Parsing Using Dynamic Time Warping.....	16
8.4.4 Testing and Results	19
8.4.5 Future Directions in Phonetic Parsing.....	21
8.5 Detection and Classification of Stop Consonants.....	21
8.5.1 Background	21
8.5.2 Features for Classifying Stop Consonants.....	23
8.5.3 Project Description.....	24
8.6 Understanding Natural Language - Cockpit Speech.....	24
8.6.1 Introduction.....	24
8.6.2 A Review of Natural Language Understanding.....	25
8.6.3 Conceptual Analysis.....	27
8.6.4 System Design and Specifications	28
8.6.5 Implementation.....	29
8.6.6 Testing.....	29
8.7 References	30

8.1 Executive Summary

The RIT NAIC project consists of the development of techniques that can be applied to speaker independent, continuous speech, large vocabulary speech understanding systems. It is our belief that Artificial Intelligence (AI) methods can provide new insight into the extremely difficult task of building these systems. This AI approach is in contrast to the traditional acoustical engineering approaches which have been used in the past.

The ultimate goal of our speech understanding research is to demonstrate an end-to-end system starting from the acoustic waveform and ending with a knowledge representation of the utterance. Such a system would provide us with a framework for both demonstrating our speech understanding techniques and comparing them with more traditional methods of speech and signal understanding. We have completed a speech and signal processing workstation which gives us the capability to assemble this end-to-end system.

Within the system there exist several milestones which represent an increase in the level of understanding along the hierarchy from acoustic waveform to speech understanding. These milestones are: (1) Derivation of the broad phonetic classes which represent the utterance, (2) derivation of the phonetic transcription of the spoken utterance, (3) the ability to map these (possibly errorful) phonetic transcriptions onto a large vocabulary, (4) a method of extracting only the plausible parsings from these transcriptions, and (5) the ability to build a knowledge representation of the utterance from a plausible parse using all possible sources of syntactic, semantic, and domain knowledge. At this time there is ongoing research at each of the above levels.

We have been attacking the broad phonetic classification problem from two fronts. The first has been statistically based classification using both K-means clustering based on Euclidian distance measures and multivariate maximum likelihood distance measures. The second approach has been based on the use of back-propagation neural networks. Both approaches train on low-level features of the signals that are most closely associated with phonetic content.

Our work in the derivation of phonetic transcriptions from broad phonetic classes is based on looking for low level features that can be used to classify a segment of known broad phonetic category into the correct phoneme. Our first research in this area was an expert system to identify fricatives. We have just begun in this past year work on classifying stop consonants. A new project, beginning in the last year of the study, will use neural network classification techniques to classify vowels from vowel-like segments.

We have made a single attempt at using Dynamic Time Warping (DTW) techniques to parse errorful phoneme strings into words from a vocabulary of approximately 800 words. We have found that this technique is computationally expensive and produces many words strings in addition to the correct string. We plan to examine some other less computationally expensive methods and compare their results.

Our work in determining plausible parsings is still under consideration. Our work in natural language understanding has progressed to the system design phase. We will be looking to apply conceptual analysis techniques to the domain of cockpit speech in fighter aircraft. This work focuses on the areas of ill-formed and ungrammatical input.

8.2 System Architecture

The architecture of the system has moved from a paper design to an actual computer system. We have completed our speech research software environment which runs on the TI Explorer and microExplorer systems. The system is called ESPRIT and will be the development, testing, and delivery bed for the speech understanding system.

ESPRIT allows us to implement our various speech understanding algorithms in the LISP environment and to build both top-down and bottom-up implementations of the system. Software that has been completed can be implemented under ESPRIT while segments of the structure which have not been completed or are under development may be stubbed or simulated or the data may be hand-massaged before passing it to a higher level piece of software.

Much of the work done in the past year has been the implementation of software running on Unix/C environments into the ESPRIT system. We have also begun coding several of the fine phonetic classifiers and linguistic modules. The phonetic string parsing system and cockpit speech understanding system are being developed on the Explorer so no porting will be necessary.

8.2.1 Software Architecture

The software architecture of the system is largely unchanged from the past two reports. The system is still a knowledge based system attempting to capture the knowledge that experts use when reading and interpreting spectrograms. Most of the low level feature extraction work has been completed and we are near completion of the next higher level, broad phonetic classification. For the sake of continuity, we will briefly review the software architecture of the system.

A digitized speech sample is processed using standard signal processing algorithms to obtain low-level features of the signal over time. These

algorithms include FFT and LPC analyses, formant and pitch tracking, energy, zero-crossings, etc. These features are then used as input to classification modules which attempt to recognize phonemes. This first transition from raw waveform to feature sets provides us with a significant reduction in data without sacrificing the knowledge necessary to perform intelligent recognition further up in the system hierarchy.

The first classification module determines broad phonetic categories. This module attempts to segment the signal into discrete segments based on the categories: vowel-like, strong fricative, weak fricative, and silence. These segments can be thought of as regions of the utterance that are roughly homogeneous, and they represent the results of the first coarse phonetic segmentation of the utterance.

These broad phonetic segments are then analyzed by modules which attempt to assign phonetic labels to the broad phonetic segments. These modules do not necessarily pick a single label, but more often they assign labels with probabilities, thus allowing higher level processes to disambiguate any inconsistencies and avoiding serious losses of data in the low level routines. This approach of assigning confidence factors to several labels for a segment parallels the manner in which human spectrogram readers perform.

Once this lattice of possible phonetic labels and probabilities has been generated a high-level module hypothesizes word candidates from a lexicon. At present we have investigated the use of Dynamic Time Warping to hypothesize words from a phonetically ordered lexicon. High level knowledge of the English language is incorporated at this level using the probabilities of two English phonemes occurring in succession and the probabilities and locations of common phonetic errors such as insertion, deletion and substitution. We are using the same lexicon for this level as we will use in the higher level natural language understanding system.

Once we have a candidate word hypothesis we will generate a single utterance from the candidate words. This selection is based on the confidence factors generated by the word hypothesizer and world knowledge available about the domain. This utterance will be considered the correct transcription of the raw signal and will be passed on to the natural language understanding system. The natural language understanding system will provide feedback to this level to resolve ambiguities or have another utterance hypothesis made if the first attempt could not be understood. This level of the system is best described as the utterance hypothesizer.

The highest level of the system is a natural language understanding system. This module will attempt to build a semantic representation of the input utterance using all possible high level knowledge sources including domain knowledge, syntactic and semantic information, and auditory cues

such as pauses and inflection. Conflicts that cannot be resolved at this level will communicate with the utterance hypothesizer to get a better representation of the spoken signal. The final output of the system will be a model which captures the intent of the spoken utterance.

The software architecture just described is primarily a data driven or forward chaining type of control strategy. This strategy is based on the assumption that a reasonably accurate phonetic transcription of the raw speech signal can be produced by the low level modules in the system.

8.2.2 Hardware Architecture

Most of the work of moving the project to the TI Explorer/Odyssey workstation has been completed. The TI Explorer is high performance Lisp Machine. It contains a central processor which uses Common Lisp as its machine language thus achieving speeds not possible with interpreted Lisp. The Odyssey board provides four TMS 32020 signal processors which can execute the traditional LPC and FFT algorithms at better than real time. These processors can operate in parallel or serially, and up to 16 boards may be installed. This provides a maximum of 64 processors all performing in parallel.

We have upgraded our two Explorer I workstations to Explorer II workstations which provided a 500% increase in the execution speed of most system functions. We have also acquired two TI microExplorers which provide Explorer I performance and capabilities on a board which drops into an Apple Macintosh II. These microExplorers have increased our development capabilities tremendously by allowing more access to the LISP environment for our staff and consultants.

The speech analysis workstation that had been under development has been completed. It is the Explorer Speech Processing workstation at Rochester Institute of Technology (ESPRIT). In the next section we will describe ESPRIT and how it is used as the hardware framework for developing and delivering the speech understanding project.

8.2.3 The ESPRIT System

ESPRIT is a speech research development environment that runs on the Texas Instruments Explorer Lisp workstation, optionally augmented with one or more Texas Instruments Odyssey Signal Processing boards. This system also will run on the TI microExplorers which do not have Odyssey boards.

ESPRIT's main goal is to provide speech scientists, linguists and engineers an intuitive software environment in which to study speech signals and to provide tools for conducting speech research. The basic functions of ESPRIT are to collect, process, and graphically display raw and

processed speech signals in ways that are useful to speech scientists. No prior knowledge of Lisp or any other programming language is necessary, and no prior knowledge of the operation of the TI Explorer is required in order to perform a wide variety of speech processing tasks.

Users may use ESPRIT interactively to perform simple operations one at a time and display the results after each operation is performed. These operations and displays include raw waveforms, FFT and LPC spectrograms, and other useful parameters and features that can be extracted from speech signals.

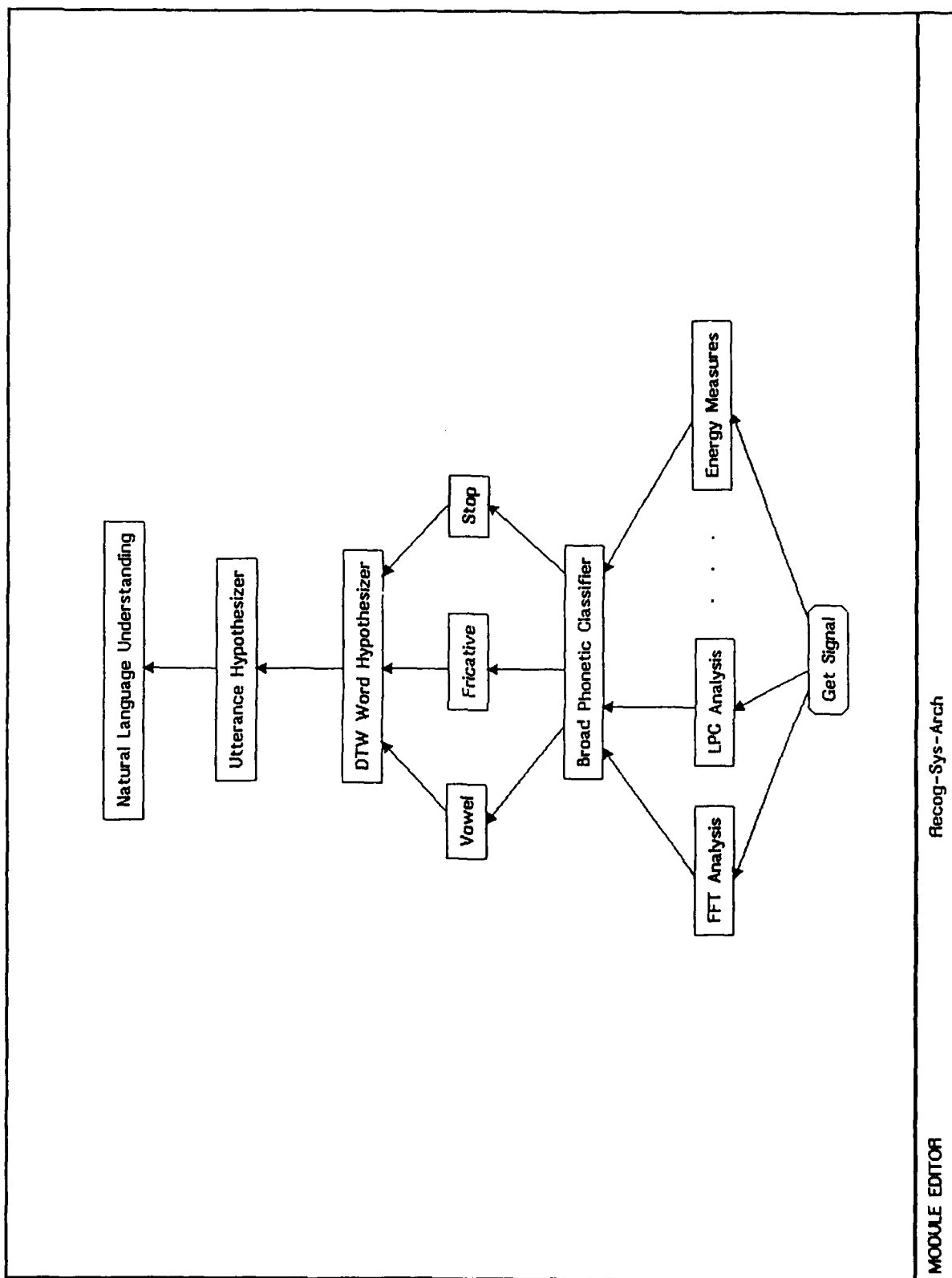
Users also may build modules made up of simpler operations and displays to perform complex tasks. This feature allows users to literally "draw" a sequence of speech processing functions and display directives and then execute the resulting "program" to perform the task that was drawn. This allows a user who is not a programmer to put together existing programs into a configuration that performs some desired task without having to type a single line of code.

The ESPRIT user interface takes a mouse-and-menu approach, and in fact, the entire system can be run by clicking the three buttons on the Explorer's mouse or by clicking the single mouse button in combination with a few keystrokes on the microExplorer. Help is available at all times for all commands, both in the form of mouse command documentation, which is always displayed automatically, and in the form of more extensive documentation, which may be displayed easily on demand.

Care was taken in the design of ESPRIT to make the displays and mouse buttons as consistent as possible. This helps the user to develop sound instincts for how to use the system and view the displays. For users who feel a need to type keystrokes instead of navigating through menus, all commands on all menus have corresponding keystroke equivalents.

Both the module building capabilities and graphical display capabilities are heavily used in implementing the speech understanding project on the Explorers. The section of ESPRIT used for building modules is the Module Editor. Through the Module Editor users can build up complex processes by describing how the pieces fit together graphically. Figure 8-1 on the next page shows the software architecture of the speech understanding system as it is built under ESPRIT. Figure 8-2 shows some of the graphs that ESPRIT can generate. These graphs allow us to evaluate the signal processing algorithms used in the system and to make exact measurements from the analyses.

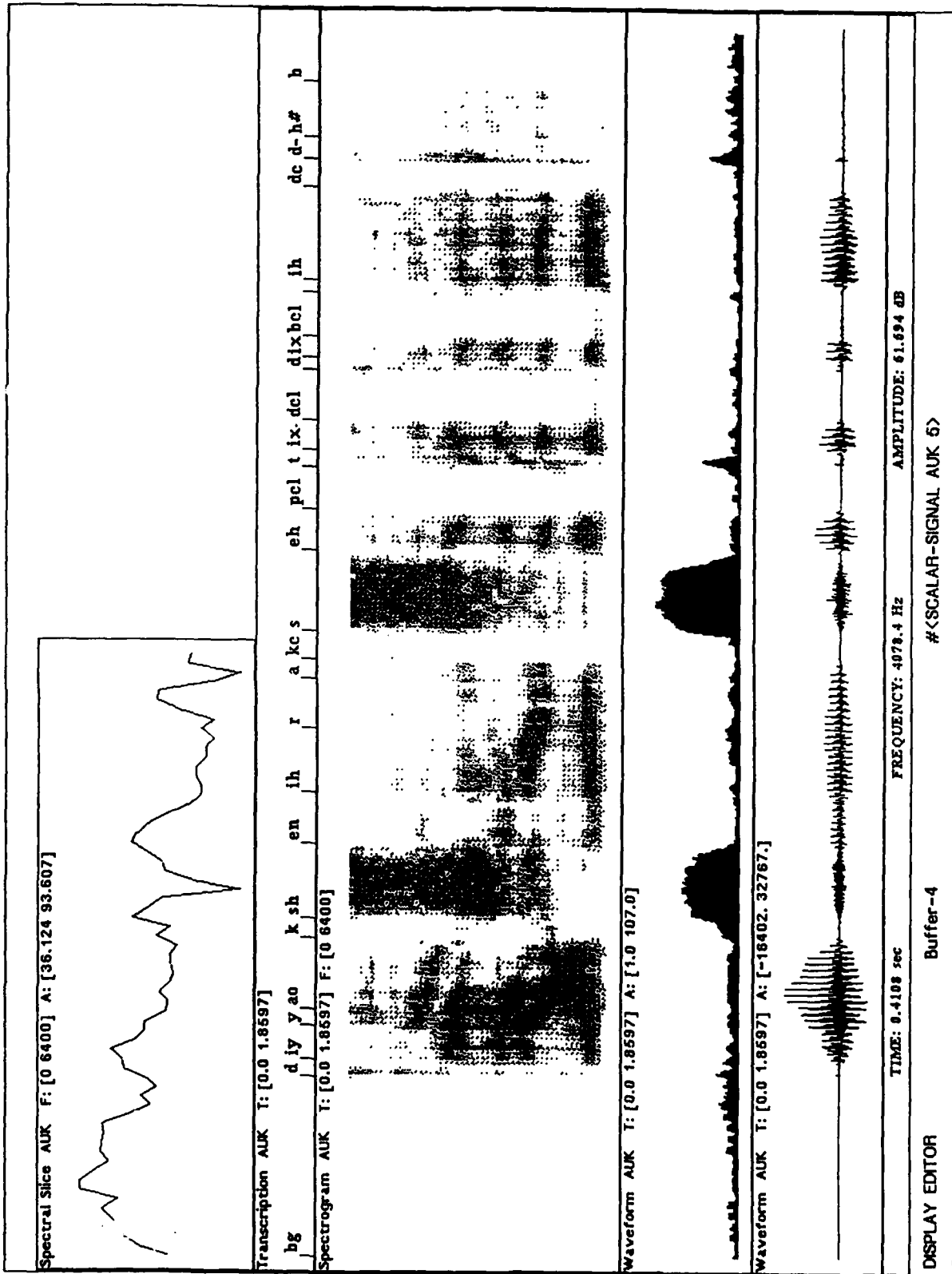
Figure 8-1 Meta-Module of Speech Understanding



Recog-Sys-Arch

MODULE EDITOR

Figure 8-2 ESPRIT's Graphical Capabilities



The ESPRIT environment itself is an object-oriented system built around the following conceptual objects: data objects, processes, displays and meta-modules. Data objects are data structures which hold raw speech, sequences of FFTs or LPC spectra, sequences of phonemes or words. These data objects may be permanently stored as files or dynamically created and destroyed throughout the execution of the user's application.

Processes are TMS 32020 code or LISP code which are used create, analyze, and destroy the various types of data objects. ESPRIT contains a large number of signal processing routines that may be used by any application. Users may also develop their own processes and incorporate them into the ESPRIT environment.

Displays are the graphical windows which are used to display and measure the data objects. Several types of displays are seen in Figure 8-2. The environment stores the knowledge to correctly display the various types of data objects, or the user can specify a different type of display other than the default.

The most important capability provided by ESPRIT is ability for users to build their own meta-modules. A meta-module is a directed graph that contains process objects, display objects and possibly other meta-modules as well. A graphical interface allows the users to draw their applications. Figure 8-1 is a meta-module which describes the speech understanding system.

8.3 Broad Phonetic Classification

8.3.1 Overview

We have completed one phase of our work in broad phonetic classification. A system known as CLASS has been developed which is able to classify segments of continuous speech as vowel-like, silence, weak fricative or strong fricative. These segments can then be passed to other more specific classification schemes under development that will attempt to identify the exact phonemes present in these segments. The CLASS system examined two classification schemes and three decision making architectures.

The two classification schemes examined were both cluster-based analyses. In the first scheme a maximum likelihood clustering algorithm was used and in the second scheme a K-means algorithm was used. CLASS used tree-based decision architectures to maximize the probability of correctly separating dissimilar classes. For example vowel-like segments would be classified on the opposite side of the tree from silent segments as they are not very similar with respect to the features generated by the signal processing feature extraction routines. Both the classification schemes and tree structures will be discussed in Sections 8.3.3 and 8.3.4.

8.3.2 Training

The training portion of classification involves the examination of many samples of speech data in order to learn the salient characteristics of the categories in the decision space. The training done with respect to broad phonetic classification involves feature extraction and cluster analysis.

In the feature extraction phase, raw samples of speech were analyzed by various signal processing algorithms to construct a data structure which describes the correct phonetic labeling of the speech segment and a vector of results from the signal processing algorithms. This data structure is known as a label/vector pair (LVP). These LVPs are then used in the clustering analysis phase as the basis for dividing up the decision space. The signal processing features used in the LVPs were zero crossing rate, total energy, relative energy, peak, and spectral change.

The speech data was taken from a data base supplied by Carnegie-Mellon University. It includes 371 utterances, of which 143 are rich in fricatives, 129 in stops, and 99 in vowels. The data base was spoken by 35 different speakers and has been hand labeled with phonetic transcriptions. Both training and testing data were sampled from this data base.

8.3.3 Cluster Analysis

Once the feature information has been extracted and the LVPs have been computed, clustering algorithms assign each LVP produced in the training data to one of the four coarse phonetic categories. The goal of this phase is to separate the LVPs into four clusters where each cluster corresponds to one of the four categories.

The decision making structure inherent in performing cluster analysis can be done in many ways. It can simply look at all the features in the LVP and make a single decision or it can be broken down into a series of smaller decisions looking at a subset of the features during each decision. The latter technique allows more control over the performance of the system by providing the ability to make decisions about highly separable classes early in the hierarchy and then make the more difficult decisions about closely related classes later. This decision hierarchy is best represented by a tree structure. The tree structures examined in the system are diagrammed in Figure 8-3.

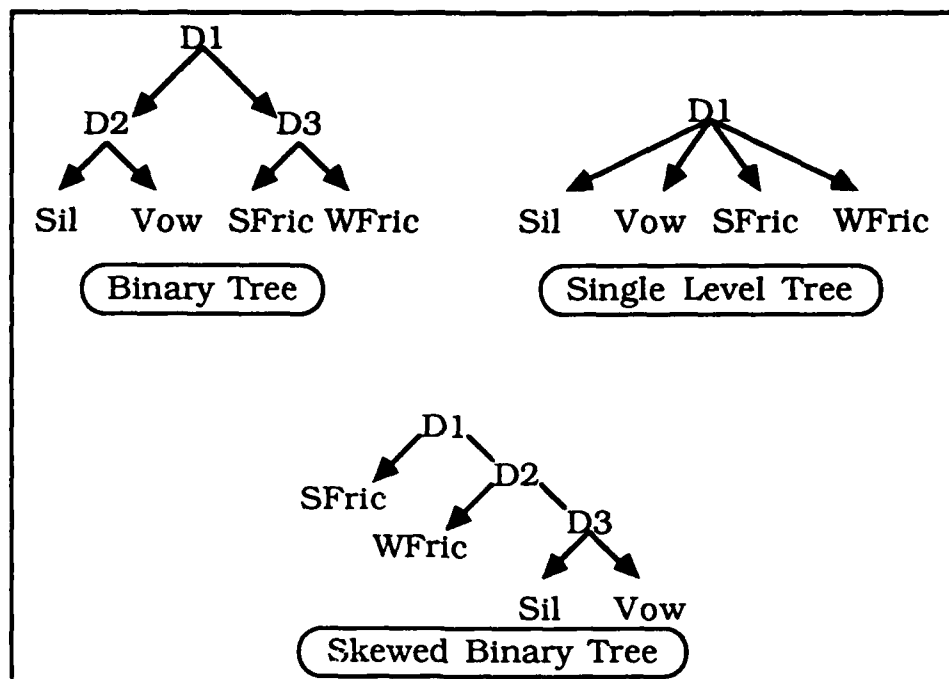


Figure 8-3

These three different decision trees can be combined with the two clustering algorithms to form six distinct methods for performing broad phonetic classification. In Figure 8-3 the nodes labeled D1, D2, and D3 represent points where a decisions are made and the input is categorized as one of the outputs (shown with arrows).

The most obvious approach to clustering is to place the LVPs into one of the four classes based on the phonetic label of LVP which came from the hand labeling in the data base. Knowledge of all the terminal nodes in the tree structure determines how the data are clustered at each decision point. The advantage of this approach is that if the phonetic label is correct, then each terminal node will be the most complete representation possible of its class with the given data. The drawback is that that if the label is incorrect, bad or misleading information will propagate through the hierarchy.

When the clusters have been determined in this manner the next task is to determine which features should be used at each decision point in the decision hierarchy. An algorithm known as maximum likelihood computes the classification based on every combination of the features. The feature set which produces the minimum number of errors with respect to the hand labeled section of the LVP. The classifications in this step are being made with a distance measure to the centers of each of the clusters.

A second approach to clustering is to look only at the feature values in the LVPs to separate the data. The K-means algorithm generates K cluster centers (means) for data points in an n-dimensional space. Each LVP is

considered to be a point in the n-dimensional space and the algorithm will find four cluster centers for this data. The inputs to the algorithm are the LVPs and K initial cluster centers. These initial centers may be arbitrary and are often the first K data points given.

The algorithm then iteratively places the remaining data points about these centers based on minimizing the Euclidian distance of the data points to the cluster centers. New cluster centers are then calculated by minimizing the the squared distances from all the points in the cluster to the center. The algorithm adjusts the centers and clusters until the center points stabilize. This algorithm makes no use of the labels in the LVPs as it does the clustering. the labels are used only for performance evaluation.

8.3.4 Results of Classifying Unknown Data Samples.

Once the training is done the classification is relatively simple. Starting at the root node in the decision hierarchy, the feature vector of the unknown speech frame is computed. The distances are calculated from the feature vector to each of the cluster centers associated with the children. The node associated with the minimum of these distances is then selected, and the process is repeated until a terminal node is reached.

8.3.4.1 K-means and Maximum Likelihood Results

In each of the results tables below there are 4 methods of measuring a correct classification: 1) percentage of correctly classified frames out of all frames; 2) percentage of correctly classified frames disregarding frames that are within 10ms of a segment edge; 3) percentage of correctly classified segments to within 10ms of the true segment edge; and 4) percentage of correctly classified segments anywhere within the true segment.

Of these measurements the first and second are the truest tests of performance. The second measurement is valid in that it has been shown that hand-labeling is accurate to only about 10ms, thus errors within 10ms of the hand-labeled boundary are not as serious a problem as errors near the centers of a segment.

K-means	1	2	3	4
Single Level	77%	84%	87%	48%
Binary	79%	84%	91%	49%
Skewed Binary	78%	83%	90%	57%

Max. Likelihood	1	2	3	4
Single Level	68%	73%	86%	46%
Binary	80%	85%	91%	58%
Skewed Binary	76%	82%	85%	54%

From these classification performance figures, the best classifier was the maximum likelihood binary tree structured system. It was the best or tied for the best in all four types of correctness measure.

8.3.5 Neural Networks for Broad Phonetic Classification

We have just started work which replaces the K-means and Maximum Likelihood clustering algorithms with Back Propagation Neural Networks at the decision points in the three decision hierarchies. This was done in attempt to see if the neural network approach might find elements in the feature set data which allow better separation of the classes than was capable using the more statistically based clustering algorithms.

Preliminary results from this work seem to indicate that the neural network approach provides similar results as the Maximum Likelihood, binary-tree type of cluster-based analysis.

We also plan to use neural networks for vowel classification over the next year of the project.

8.4 Phonetic String Parsing

8.4.1 Background

As the result of previous processing on several levels, an unknown utterance is transformed into a string of undifferentiated phonemes (i.e. no word boundary markers). The phonetic string parser scans this sequence of phonemes, hypothesizing all words in the utterance that are consistent with the lexicon. Generally, this can be accomplished by comparing reference patterns in a phonetic lexicon with the unknown sequence. All words hypothesized are then passed to a syntactic and semantic parser for further analysis.

One needs to consider the potential problem of a single phonetic string with multiple interpretations as a sentence. Ambiguous parsings can map a single phoneme sequence into different strings of words. Matching against entries in the lexicon will not contribute to solving this problem. There is a need for syntax and semantic knowledge to differentiate meanings. This is outside the scope of the phonetic string parser. What is required to solve this problem is feedback from the Natural Language Understanding and Utterance Hypothesizing levels of the system. We understand the important role that this feedback plays, but until we have investigated the system without feedback we can't correctly design this top down integration with the high level modules.

There are three areas of complexity which prevent lexical access from being a simple lexicon look up. Front-end errors, and the effects of phonological recoding, are two areas which alter the symbolic

representation of an utterance. Ambiguity that results in multiple parsings from a single phonetic representation is the third area.

It has been observed that the front end of a speech understanding system will at times exhibit an inability to distinguish between similar sounding phonemes. As a result of this confusability, the string of phonemes representing an unknown utterance may contain insertion, deletion, or substitution errors. [COHE75], [KLAT75], and [OSIK75], show that in continuous speech there are variations in pronunciation (especially across word boundaries) that are not random and can be described by a set of phonological rules. Finding lexical search methods that compensate for phonological recoding has not proven to be a simple matter. Phonological rules that apply within a word boundary can be handled by creating an alternative base form (i.e. an idealized pronunciation) entry in the lexicon. However, a more difficult problem arises when working with variations across word boundaries. For example, the utterance "did you" may actually be realized as "di-ja", illustrating a deletion and insertion error. Adding another base form to the lexicon for the word "you" (starting with the palatal "j"), could provide for an erroneous recognition of "you", when the utterance may in fact be "judge". Even with this potential problem, some recognition systems such as HWIM [WOLF77], SPEECHLIS [WOOD75], and the lexical access system at Carnegie-Mellon [RUDN87], have built lexicons where each word may have alternative representations. These are generated by application of phonological rules to dictionary base form representations.

8.4.2 Approaches To String Comparison

The two most commonly cited approaches to string comparison in isolated speech recognition are Hidden Markov Models [LEVI83] and Dynamic Time Warping [ITAK75]. Both methods operate on the general principle of dynamic programming in searching for optimal paths. Although both approaches have been extended into the domain of continuous word recognition ([BAKE75], [WOLF77], [LOWE80], [MYER81], [NEY 84], [LEVI87]), it is not certain if they are extendible to large vocabulary, speaker independent, continuous speech understanding systems. Neural Networks have also been used to examine the phonetic parsing problem as well as the relatively new field genetic algorithms.

The use of Hidden Markov Models (HMM) are used to model a stochastic processes (Markov sources) that are not directly observable, but can be examined through the output of sequences of symbols [RABI86]. Phonemes and words can be thought of as the observed output dependent on the probabilistic changes (transitions) in acoustic signals and phonemes respectively. Through the use of training utterances and empirical observation of a system's front-end performance, one can model the process of phoneme and word generation, taking into account front-end errors, speaker variation, coarticulatory and phonological recoding effects. Words within the unknown utterance would be hypothesized as those whose

models had the greatest probability of generating the observed phonemes. The Dragon system [BAKE75], incorporated the concept of chaining Markov processes in a hierarchical fashion, not only on a word basis, but at the phrase and sentence level as well. The result was a finite state network of Markov sources in which the recognition procedure looked for an optimal path of transitions that would most likely account for the observed utterance.

Dynamic Time Warping (DTW) is a method of sequence comparison derived from a time sampling of some quantity that is subject to variations. DTW has been successfully used in isolated word recognition by [ITAK75] and [WAIB81], in addition to connected word recognition by ([MYER81], [NEY 84], and [WATA86].)

An unknown utterance and a reference utterance can be thought of as two sequences of feature vectors or tokens (phonemes). Each sequence defines the axis of a matrix mapping the speech utterances against one another. At each coordinate is a measure of distance or dissimilarity between the tokens. The goal is to find a path between endpoints of the two sequences whose distance D is minimal. This cumulative distance can then be used as a decision criterion for recognition [NEY 84].

Applying the principle of optimization from dynamic programming concepts [NEY 84], a recurrence relation minimizing the number of points considered at any one time follows:

$$D(a_i, b_j) = \min \left\{ \begin{array}{ll} d(a_{i-1}, b_j) + w(a_i, 0) & \text{deletion of } a_i \\ d(a_{i-1}, b_{j-1}) + w(a_i, b_j) & \text{substitution of } a_i \text{ by } b_j \\ d(a_i, b_{j-1}) + w(0, b_j) & \text{insertion of } b_j \end{array} \right\}$$

The equation above generates the distance between any two endpoints and at the same time minimizes the number of points that are examined.

Weighting coefficients are added to penalize for deletions, substitutions and insertions. However, searching all possible paths is computationally expensive. Constraints that restrict this search include controlling the degree of slope allowed in the warp, and setting some maximum permissible path distance. These will prune paths that would otherwise grow excessively large.

Hidden Markov Models require the collection of empirical statistics that describe the response of the recognition system's front-end. The determination of states, transitions, and associated probabilities is a complex optimization problem. More significant is the fact that our front-end is not complete, precluding any statistical evaluation. Dynamic Time Warping, on the other hand, needs only the strings for comparison and the

provision of some distance metric. The drawback to DTW is that it requires a large number of distance calculations. A study by [LEVI83] estimated that HMM, which uses a simpler likelihood evaluation function, requires an order of magnitude less computation time than DTW. Also noted in this study was that both systems achieved comparable error rates.

8.4.3 String Parsing Using Dynamic Time Warping

Central to the method of DTW is the necessity of some distance measure. Once this measure has been determined, the process of sequence comparison can proceed. Studies by Miller and Nicely [MILL55] demonstrated that humans typically confuse particular consonants in a consistent fashion. Predictable confusability patterns are also exhibited by the acoustic-phonetic modules of speech recognition systems. Ideally, we would make use of the front-end's response characteristics in classifying all phonemes as a distance measure. However, the vowel classification study by Hillenbrand and Gayvert [HILL87] is the only portion of the front-end for which data exists. The remainder of the data had to be extracted from studies of human confusability by Shepard [SHEP80].

The lexical knowledge source for this study is the vocabulary taken from a United States Air Force Cockpit Natural Language study by Lizza et al., [LIZZ87]. The study provides a vocabulary of 656 words, their frequency of occurrence, contextual use, and the number of times a word is preceded or succeeded by other words. This information may be valuable in determining the types of contextual effects to expect. Using a text-to-speech synthesis system (DECtalk - manufactured by Digital Equipment Corporation) in combination with hand coding, phonetic transcriptions have been produced for all words in the vocabulary.

Similar to other level building algorithms (Sakoe and Chiba, [SAKO79] Myers et al., [MYER81], Ney, [NEY 84],) the DTW procedure moves from left to right, finding the collection of reference patterns whose global (phrase) DTW distance is at a minimum over the concatenation of local (word) DTW minimums. A phoneme reference pattern is selected from the lexicon (the selection method will be discussed below) and time-warped with an initial portion of the unknown utterance, producing a time-normalized distance. This procedure is applied repeatedly to the same section of the unknown, until all acceptable word hypotheses are determined. Hypotheses that exceed a preset minimum distance threshold during DTW calculations, or deviate from other constraints are pruned early. This reduces the expenditure of computational resources by eliminating otherwise wasteful calculations. Once a set of hypotheses is generated, the position in the unknown utterance corresponding to just after the end of each hypothesized reference, becomes a starting point for continued DTW analysis. The process repeats until the end of the utterance is reached leaving a collection of possible word sequences.

Note that, given an utterance of fixed length, and given an equivalent distance between all reference and unknown patterns, a small number of large words will have less total accumulated distance (globally) than a larger number of small words. This indicates a possible heuristic that favors use of large reference patterns for DTW prior to smaller patterns. Smith et al. [SMIT80], suggested that large words should be hypothesized prior to smaller words since larger words usually contain more syntactic and semantic value.

Exhaustive search of all lexical entries is not practical with lexicons numbering in the tens of thousands. However, the front-end of our system is not complete, and a benchmark of system performance was to be established. Therefore, a brute force search technique was initially implemented, allowing an examination of the algorithms response relative to insertion, deletion and substitution errors. Parsing times for errorless strings was unexpectedly high at approximately ten seconds per sequence, with a sequence containing an average of fifteen phonemes. When the same set of strings with five percent substitution errors was tested, the times increased substantially, from a total time of twenty-four minutes to over one hour (with only 53 percent of the test complete).

Two steps were taken to help constrain the number of reference patterns to be used for DTW and increase general performance. First, reference patterns of the lexicon were organized as a hash table, indexed by the first phoneme that could be encountered in all reference representations. During DTW, the first phoneme in the unknown pattern provides an index to reference patterns starting with that particular phoneme (i.e. direct access to a subset of candidates most likely to match). This method relies on the premise that the first phoneme in the unknown can be identified accurately. Another assumption is that as hypothesized portions are parsed from the unknown sequence, accurate word boundaries are realized. However, due to probable insertion, deletion, and substitution errors, the locations of word boundaries are not certain. Second, due to the recursive nature of the parsing procedure, alternative parse points would cause the parse procedure to examine the same sections of phonemes repetitively. Using a global structure, parsings from specific segments were stored such that if a new parsing was to begin at the same index, the previously found parsing would be immediately available. This reduced the parsing times for errorless strings by approximately fifty percent. The impact on errorful phrases was not as great.

Sakoe and Chiba ([SAKO78]) detail five general conditions that typically restrict the warping function. The first two are that the function be monotonic and continuous. Phonemes in the reference and unknown patterns are assumed to be time-ordered with their intervals relatively uniform, satisfying the first two conditions. The three remaining conditions (boundary, adjustment window, and slope constraint) are variable and can affect the relative performance of the warping procedure. So far, these

three conditions have been held constant until an initial evaluation is performed on the most basic DTW algorithm and search procedures.

Boundary conditions (i.e. sequence endpoints) are fully known for both the reference and unknown patterns in isolated word recognition. However in continuous speech, endpoints (at the word level) in the unknown utterance are not fully established, and can be highly variable in number and position. Mapping a single reference word to a disproportionately long phrase would produce an unrealistic correspondence, in addition to wasting computational resources. Therefore, some criteria must be established for selecting the appropriate length of the unknown sequence for DTW comparison. Assuming the front-end's error rate is below 100 percent, one can hypothesize that there is a maximum number of phonemes (including errors) in the unknown pattern which must be examined in order to find a word, or exhaust all possibilities. This value would be equal to the phoneme count of the reference word, plus a buffer to allow for insertion errors that can extend the unknown sequence. This buffer value was chosen to be the same number of phonemes as the reference pattern.

The adjustment window and slope constraint conditions affect the degree in which the DTW procedure accepts insertion and deletion errors. When finding a least cost path through the distance matrix, the warping path will cut a diagonal line with a slope of one if both patterns are time aligned. Deviation from the diagonal indicate larger the difference between the two patterns. Excessively long horizontal or vertical paths indicate that unusual expansion or compression is required to match two patterns. The adjustment window forms a diagonal corridor somewhat parallel to the warping function. This "window" constant has the effect of limiting the number of acceptable insertion and deletion errors.

Kruskal and Sankoff [KRUS83], Myers et al.[MYER81], and Sakoe et al. [SAKO78], demonstrated the use of slope constraints which defined a parallelogram surrounding an optimal (diagonal) warping path. Calculations that result in a path that extends beyond this boundary are terminated, constraining the number of insertion and deletion that would otherwise result in excessively long paths. It also precludes wasteful calculations. Sakoe and Chiba's [SAKO78] study showed that optimum DTW performance was maximized at a slope value of 1 in a range from 0.5 to 2. This slope value built into the equation, results in the following recurrence relation to be initially used for the DTW procedure:

$$D(a_i, b_j) = \min \begin{cases} d(a_{i-1}, b_{j-2}) + d(a_i, b_{j-1}) + d(a_i, b_j) \\ d(a_{i-1}, b_{j-1}) \\ d(a_{i-2}, b_{j-1}) + d(a_{i-1}, b_j) + d(a_i, b_j) \end{cases}$$

The equation above modifies the previous distance measure to incorporate the parallelogram surrounding the optimal warping path.

Averaging the total accumulated distance over the reference pattern length is used to normalize distances between hypotheses whose reference patterns differ in length. This also favors a heuristic that looks for the longest pattern with minimum distance. All the conditions and variables above can be adjusted to optimize the recognition procedure.

8.4.4 Testing and Results

As previously mentioned, the system front-end is not complete and requires that we simulate it through the use of human confusability and vowel classification studies. Test data and error conditions must be simulated based on those studies as well. Developing a confusion matrix for generation of simulated errors required establishing a relationship between perceptual distance and confusability both for Hillenbrand's and Shepard's data. Initially, a logarithmic relation was tried. What resulted was confusion probabilities that were unrealistically low. After a number of tries, an inverse relation of a phoneme's distance to all other phonemes proved to be adequate. Additionally, the following assumptions were made due to lack of data: (1) vowels and consonants (other than liquids or glides) are not confused; (2) diphthongs and syllabic resonants were not included, but were represented in a word by their component phonemes.

Generation of test phrases is as follows. While advancing through the input string, a random number generator selects the type of error (substitution, deletion, or insertion) to occur at a given phoneme in the string. Though assumed independent, frequency of occurrence favored substitution errors after studies by Jelinek (1976) and Ohguro (1988) suggested substitution errors account for between sixty and ninety percent of the errors.

Since many factors (as described above) influence the performance of the DTW algorithm, our initial approach was to use a simple, brute force search technique, comparing all lexical entries to error-free unknown sequence portions. As was described above, initial testing was slow in the extreme. Searching the entire lexicon became prohibitive even with a lexicon of only six hundred words. Not only was execution speed slow (on the order of hours), but memory requirements were large. The algorithm was changed so that the phoneme under consideration in the unknown hashes to the group of reference words starting with that phoneme. This helped reduce the number of candidate reference patterns, but allowed a missed

hypothesis should the first phoneme be an error. The following example serves to illustrate a few of the problems encountered so far. The phrase "time and location of rendezvous" is represented phonetically as:

t aa ih m ae n d l ow k eh ih sh ax n ax v r aa n d ix v uw

With the following errorful and equivalent parsing...

sh aa ih m ae n dx l aa sh ih sh ax n ax v r aa n d ix v uw

eye man on of rendezvous

An attempt to locate the word "time" from the starting phoneme is not possible if one only considers candidates starting with "sh". In this case, reducing the candidate search space helped speed processing, but missed a valid hypothesis. Current testing is attempting to compromise. So as not to become too restrictive, the sets of reference patterns selected for DTW comparison are based on the actual phoneme in the unknown, and those that are the most highly confused with the unknown's phoneme.

Another problem encountered is that the lexicon for this application consists of words whose phoneme count is very small. This has made adjusting the threshold value quite difficult. With a low threshold value, many of the larger word candidates are eliminated. Conversely, if the threshold is too high, many false positives are hypothesized. This in turn extends the search time. The algorithm now uses two different threshold criteria dependent on phoneme count of the reference word.

Early algorithms would advance over large portions of the unknown string if they were not able to find any word hypotheses. In an effort to find all possible words, two changes to the parsing algorithm have been made (1) addition of a dynamically changing threshold value; (2) a backtracking methodology. Now when the parsing procedure moves left to right, if it cannot locate any words, it backtracks, and resumes its forward direction with an increased threshold value. This allows more errorful portions to pass the DTW comparison and produce a correct hypothesis.

As would be expected, the larger the degree of error present within a phrase, the lower the performance of the method. A set of fifty-one phrases has been subjected to errorful conditions ranging from ten to twenty percent. Performance was extremely low with no full phrase completions. The number of original words found ranged from seventy-five percent (ten percent error level) to fifty percent (twenty percent error level). The number of false positives (words found but not belonging) also increased substantially as the threshold increased. This too caused processing times to increase substantially due to more parsings.

8.4.5 Future Directions in Phonetic Parsing

Currently the effort is to improve upon the recognition of entire phrases. Though complex, it seems that the subject area that requires more investigation is the sensitivity in how the threshold value causes the acceptance or rejection of the hypothesis during the DTW procedure. Also requiring further research is the method used to intelligently select word candidates. Testing and investigation of how to adjust and use the threshold value are continuing. We will also be investigating the alternative HMM and Neural Network approaches to this problem. We also plan to investigate the feedback issue as it applies to DTW in more detail.

8.5 Detection and Classification of Stop Consonants

8.5.1 Background

We have just begun work on the acoustic-phonetic classification of stop consonants in the speaker independent continuous speech environment. The stop consonants we will be examining are /p,t,k,b,d,g/. Acoustically, a stop appears as a short period of silence followed by an abrupt release. In conversational English, stop consonants account for over 17% of all phonetic occurrences [MINE78]. Therefore, definitive classification of stops over all phonetic boundaries becomes a significant element of a complete recognition system.

Present recognition systems are divided into two categories: template matching and acoustically based segmentation and labeling. Template matching is done by parametrically representing an unknown word and comparing it to a library of known words. Template matching is not a viable approach for a complete or large word recognition system. A system of this configuration would be extremely slow since the English language has approximately 300,000 words and approximately 60,000 syllables. The second category, segmentation and labeling, attempts to recognize phonemes, which are the smallest contrasting speech sounds for a language. English has approximately 45 phonemes. Theoretically a system which classifies phonemes would be much more robust than a template based system. Thus the purpose of this research is the phonetic classification of stop consonants.

There are two ways to address the problem. The first is to model the human auditory system, which has been proven to be a high performance speech recognition system. However, the human auditory system is a complex biological machine which is not fully understood. Therefore, most research has explored data analysis techniques. The data analysis approach may consist of any number of numerical recipes aimed at sub-dividing the various classes of speech down to the phoneme unit.

Initial work developed around the idea that classification information could be found in the acoustic speech signal. The implementation of mathematical processing, e.g., Fourier analysis, linear prediction, etc., allowed for a visual spectral representation of speech over time. It was difficult to make classifications based only on these spectra due to the great diversity of the spectral shapes of a certain stop consonant over several phonetic environments.

As analytical techniques have not yet been successful, speech perception theories raise the question of how the acoustical input to the ear is translated by the brain into speech sounds. Some researchers, e.g., Allen [ALLE80],[ALLE85], believe that to implement a hearing machine one must first model the human auditory signal processing system, i.e., model the cochlea. The literature is full of experimental data surrounding the signal processing of the cochlea. From this data, Allen and other researchers (Lyon [LYON82],[LYON83],[LYON84], Seneff [SENE88], Monderer and Lazar [MOND87],[MOND88], and Ghitza [GHIT88]) have generated cochlear models. The models transform the signal to correspond with experimental data taken from the ear. Auditory models have a tendency to accentuate energy and frequency changes. They also appear to be much more resistant to noisy input, i.e., they filter out noise or non-speech components. The output from these models supports the idea that the brain makes a fundamental distinction between speech sounds and non-speech sounds [PARS87]. The cochlear model output constitutes a multitude of channels representing auditory neural transmitters. From this point on the rest of the system appears as a block box.

Thus we have disadvantages with both signal processing systems. Acoustic signal analysis generates overlapping parametric data (this is not to say that auditory models do not do the same), and the difficulty with auditory models is that the black box classification is not understood. In either case, the fundamental approach should be to capture input, process input, and classify. This classification problem becomes very large when speaker independence is considered. As a further testimony to the problem, Edwards [EDWA81] found that with 17 of the most conspicuous features used in stop sound research, a decision based on a posterior probability showed that none were sufficient, by themselves, for identification, and that combinations of features only offered redundant information.

The input for both auditory and data analysis models is the acoustic signal, which is composed of the speech, background noise, and reverberation. One aspect (for speaker independence) of generating a successful classification system is to consider a data normalization scheme. Seneff [SENE87] showed that speaker normalization for vowel classification can be obtained by pitch subtraction, (i.e., over some time interval, find the average fundamental frequency and subtract it from the formant frequency). But this approach would fail for stops since resonant frequencies are not present. However, in recent work by Yoder and Jamieson, speaker

independence is addressed by attempting to normalize the acoustic signal via the frequency domain [YODE86], [YODE87]. Yoder and Jamieson showed improved classification performance by using a hybrid of the Mellin and Fourier transforms. The primary attribute of the Mellin transform is its scale invariance, analogous to the shift invariance of the Fourier transform.

A primary question to consider is whether the noise filtering and signal enhancement qualities of the human auditory models will outperform an analytical system aimed at input signal normalization. Although there is evidence to suggest that the signal enhancement features of an auditory model will aid in the identification of place of articulation, research constraints of time and scope will limit the work in this area.

As a result of Edwards findings on the usefulness of so called "predominant features", we believe that the elusive signature in the analog sampled waveform may be better represented by a mathematical characterization, e.g., by the moments of a distribution [FORR87].

In summary, the method proposed here will be input normalization, via the Fourier-Mellin transforms, abstract spectral feature extraction and compression for input processing, and classification. The classification method will use a neural network. We expect that our previous experience in using these networks in vowel classification, phoneme parsing, and broad phonetic classification, will allow us to design a network that makes best use of the abstract spectral features discussed above. The majority of our work in this area, during the past year, has been concentrated on finding the correct features to be fed to the network and not the building of the network itself, hence a discussion at some length of the characteristics of stop consonants.

8.5.2 Features for Classifying Stop Consonants

The features we used will be derived from running spectra. This will be done because we are interested in the spectral envelope as it changes over very short segments of time. For each spectra, we want to extract features describing its structure, i.e., its spectral signature. The following have been tentatively chosen for this task:

- center frequency
- (center of mass)
- mean amplitude (distributions central value)
- mean absolute deviation (variability around the mean)
- skewness (characterized "width" or "variability" around the mean)
- kurtosis (peakedness of the distribution)

8.5.3 Project Description

The data files used will be those from the Carnegie-Mellon University (CMU) speech database. Discrete stop consonant tokens (data files) have been extracted from the individual utterances. This was done by first reading the CMU label files for start and stop times of stop sounds. Every segment of speech in the CMU label files was hand marked and classified. In cases where a stop closure preceded the stop release, the closure and release were combined to form a single stop token. The stop token label files were formatted to include a start, stop, and segmentation time (in milliseconds) and a token label to include the speaker initials, utterance identification, stop type, and indexed suffix for the stop number of that utterance, e.g., 536 599 570 fdmv4-1.t-h.3. This token file labeling uniquely defines each stop sound.

A data directory was created to hold the binary stop token files for each speaker. A C program reads the speaker-utterance stop label files and systematically creates binary token files by accessing the utterance file in the CMU database. The data directory is a centralized location for all stop token data files while the speaker-utterance label files are located in an adjacent directory.

The UNIX 'awk' filter was used to perform several analytical functions pertaining to the number of stops in a given class, max, min, and average times per class, number of sub-types in each class, etc., etc.. In addition, 'awk' has been very helpful in customizing files for batch execution of programs applied to the database.

To subdivide the problem into two groups, an attempt will be made to detect voiced vs. unvoiced stops. Searle et al., [SEAR79], claimed near 100% separation using an algorithm which detected an energy threshold change in time in a specified frequency range. Although Searle used a 3rd octave 18 channel filter bank to perform their voicing criteria on, Edwards found that a simple measure of VOT accurately separated these classes 97% of the time. If place information was available, the separation accuracy was 99%. For the present work, the CMU segmentation markers will provide the voice onset time. If time allows, a comparison between the subdivided and non-subdivided classification will be done.

8.6 Understanding Natural Language - Cockpit Speech

8.6.1 Introduction

A speech understanding system [ALLE87, WINO83] is a computer system which takes as input the sound patterns that make up speech and produces as output a model of the concepts and ideas expressed in the input. When the speech that is being understood is a type of Natural Language, the

language that evolved for human use, then speech understanding system becomes a type of Natural Language Understanding system.

The system described here is designed to understand the spoken language used by fighter aircraft pilots to communicate during flight operations [LIZZ87]. It is a challenging domain due to the unique style of communication that fighter pilots have evolved. The utterances are brief but high in information content. There is little regard to correct English grammar. Few connectives and qualifiers are used. Similar Navy transcripts have been labeled "scruffy" by Richard Granger [GRAN83]. Understanding of this language is aided by strong dependence on context, expectations, shared knowledge, and the flight experience of the pilots.

The next section will address some of the methods currently used in natural language understanding and introduce the methods used by the system. Following that will be a discussion of system specifications, implementation, and testing.

8.6.2 A Review of Natural Language Understanding

There is a great diversity in the approaches taken to understand natural language. This diversity is a result of the many different backgrounds of the people doing research in the area. These disciplines include: Linguistics, Speech Perception, Artificial Intelligence, Formal Grammar Theory, and Sociology. As a result of this diversity, the literature discussing natural language understanding is large and often difficult to search effectively. James Allen's *Natural Language Understanding* provides an excellent summary of the work that has been done in this area. Once the literature has been reviewed a general design which most speech understanding systems have followed becomes evident. This basic design for speech understanding is diagramed in Figure 8-4.

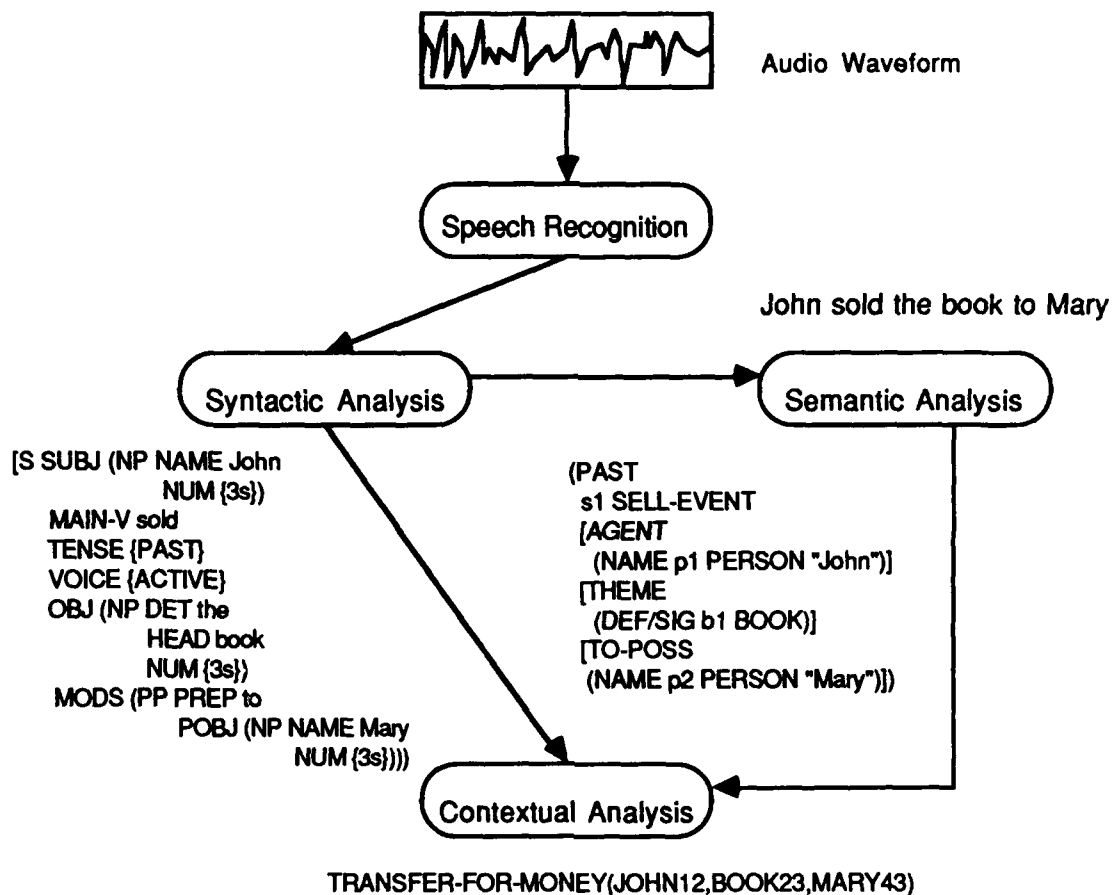


Figure 8-4

From the audio waveform a speech recognition system determines the words that make up the utterance. This system has been oversimplified in Figure 8-4 since at this level we are primarily interested in speech understanding and not recognition. The main concern in choosing a speech recognition system is to find a system which is highly accurate within the domain's normal vocabulary.

The utterance is then given to a syntactic parser which attempts to determine how the utterance fits into the grammar being used. Almost all parsers of English look at noun and prepositional phrases and are designed to parse basically well formed utterances. The text just below the Syntactic Analysis module in Figure 8-4 is the syntactic parse of "John sold the book to Mary."

This parse of the utterance then provides the semantic analyzer with the verbs, objects and phrases necessary to determine the basic meaning of the utterance. At this time both the semantic and syntactic models are passed to the contextual analyzer which attempts to resolve syntactic and semantic ambiguity, performs some pragmatic analysis to determine appropriateness, and finally asserts a representation of meaning of the utterance. Note that this representation is a result of not only the given utterance but contextual

and world knowledge which provides the environment within which the meaning is determined.

This is the basic methodology for speech understanding systems. The many systems that have been developed are modifications or enhancements on this basic design. Some of these enhancements include: better parsing techniques; a variety of semantic parsing algorithms; two-way interaction between the syntactic and semantic parsing to resolve ambiguity; methods of using information about discourse, world knowledge and context to build an "environment" for meaning.

The basic design discussed above does not apply particularly well to the cockpit-speech domain. The major problem is that the cockpit-speech is grammatically ill-formed. This makes parsing both a difficult problem and not very useful. The reason for doing a syntactic parse is that the structure of the utterance indicates the roles of the words being used. Subsequent analysis of these roles allows us to then form a semantic interpretation. If a syntactic parse doesn't give an indication of these roles than there is little point in doing one. This is the case with the analysis of cockpit-speech.

It is clear that our ability as humans to understand language is extremely robust. We are able to handle many types of errors including omissions and ill-formed input and as Granger states, "get right to the meaning" of the text. The work done by Schank and Reisbeck at Yale was developed with this goal in mind. Schank and Reisbeck introduced the Conceptual Analysis (CA) method of semantic processing which begins parsing the utterance based on the action/event indicated by the words in the utterance. Schank and Reisbeck state "meaning was the primary issue, and the study of syntax should be guided by the demands of a theory of understanding." [SCHA81].

8.6.3 Conceptual Analysis

The basic theory behind CA is that many words, especially verbs, identify case-based semantic structures. The semantic parser's job is to then fill in the values of these cases from the rest of the sentence. For example, the utterance "John sold Mary the book" implies the action of selling, the object is a book, the actor is John, and the direction of the event is toward Mary. The system is based on lexically driven pattern-action routines which allow the parser to fill in the various actors, objects, actions and directions.

There are several reasons that Conceptual Analysis fits the domain of cockpit-speech particularly well. The analyzer is lexically based and no grammatical parse is done. The utterances in cockpit speech tend to be short and simple, expressing a single uncomplicated concept which is more easily handled by the CA analyzer. The problems with the CA approach are due primarily to its lexically based approach. Extensions to the vocabulary necessitate new pattern-action routines. Adding a single new verb may not only require a new pattern-action routine, but could require several new

routines to handle the interaction of the new routine with the already defined pattern-action space.

8.6.4 System Design and Specifications

The design of this understanding system is built with the intent of providing a tunable framework for demonstrating how CA works as a semantic analysis in the domain of cockpit speech. Our hope is that by refining the pattern-action routines and tuning the situational scripts we can provide a system which is good at understanding the cockpit speech collected by Lizza et al. The following diagram represents the high level design of the system.

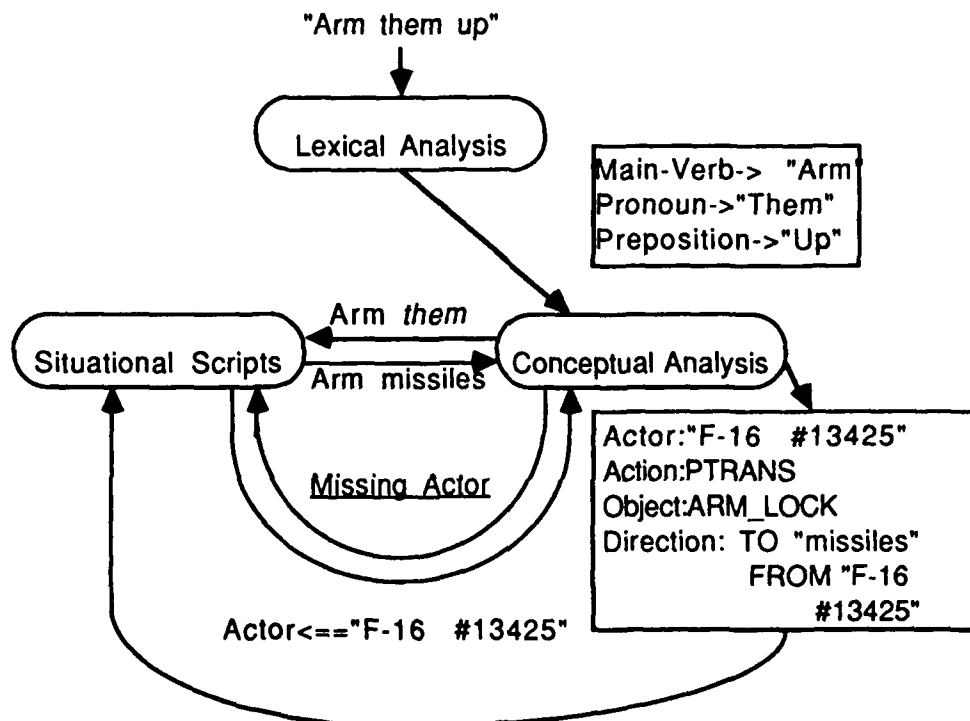


Figure 8-5.

In figure 8-5 the interaction between the situational scripts and conceptual analyzer allows the analyzer to fill in the values in the primitive action frame PTRANS (Physical Transaction.) PTRANS was chosen because conceptually arming/locking missiles can be thought of as physically placing a lock on the missiles which changes the attribute of the missiles. This is similar to the way that locking a door changes the physical properties of the door.

The choice of PTRANS and the slots to fill within PTRANS is made by the pattern-action routine that is based on the verb "arm". By altering these pattern-action routines we can alter the behavior of the system to respond better to the input. The danger here lies in writing pattern-action routines

which map every utterance in the domain. The resolution for this is to try to develop a minimum of pattern-action routines necessary to represent the concepts of the domain and also to train and test on different utterances. There is also some control in the situational scripts which to control the entry and exit from one script to another and the inferring power of the script.

In terms of the design specifications, the system is built such that it takes corrected input utterances, i.e., what was spoken is what is input. The system is said to have correctly understood the utterance if it produces a semantic representation consistent with both the input and context and produces the same semantic representation for paraphrases of the utterance.

8.6.5 Implementation

The system is implemented in Common Lisp with Flavors on the TI Explorer System. The Conceptual Analyzer is based on the work of Birnbaum and Selfridge [BIRN81] and Reisbeck and Schank [REIS76, SCHA81.] The scripts are based on the work of Schank and Abelson [SCHA77.] The training and test utterances come from the cockpit-speech study mentioned in the introduction [LIZZ87.]

8.6.6 Testing

The system will be tested on utterances not used in the building and tuning of the understanding system. For each situation in the cockpit-speech study there are approximately 40 utterances. Approximately half of these utterances will be used in building the system and the other half will be used to test the final version. None of this testing data will be used to further tune the system so as to have a good measure of the system's accuracy. An overall accuracy rate will be generated based on the correct analyses of the utterances and some of error analysis to show where the system's frailties lay.

8.7 References

- [ALLE80] Allen, J.B., "Cochlear Modeling", ICASSP, 1980 766-760.
- [ALLE85] Allen, J.B., "Cochlear Modeling", IEEE ASSP Magazine, 1985, 3-29.
- [ALLE87] Allen, James., *Natural Language Understanding*, Benjamin/Cummings, Menlo Park, California, 1987, 1.
- [BAKE75] Baker, J.K., "The DRAGON System - An Overview", IEEE Transactions on Acoustics, Speech, and Signal Processing, 23, 1975, 24-29.
- [COHE74] Cohen, P.S. and Mercer, R.L., "The Phonological Component of an Automatic Speech-Recognition System", in *Speech Recognition: Invited Papers Of The 1974 IEEE Symposium*, Reddy, D.R. ed., Academic Press, New York, 1975, 275-320.
- [EDWA81] Edwards, T.J., "Multiple features analysis of intervocalic English plosives", *Journal of the Acoustical Society of America*, Vol 69 No 2, Feb 1981, 535-547.
- [FORR87] Forrest, K., Weismer, G., Milenkovic, P.M., and Dougall, R.N., "Statistical Analysis of word-initial voiceless obstruents: preliminary data", Submitted to *Journal of the Acoustical Society of America*, 1987.
- [GHIT88] Ghitza, O., "Auditory Neural Feedback as a Basis For Speech Processing", ICASSP, vol 1, 1988, 91-94.
- [GRAN83] Granger, Richard H., "The NOMAD System: Expectation-Based Detection and Correction of Errors during Understanding of Syntactically and Syntactically Ill-Formed Text", *American Journal for Computational Linguistics*, 7(3-4): 188-196.
- [HILL87] Hillenbrand, J. and Gayvert, R.T., "Speaker-Independent Vowel Classification Based on Fundamental Frequency and Formant Frequencies", *Journal of the Acoustical Society of America*, Spring 1987, 81 (Suppl. 1), S93 (A).
- [HUTT84] Huttenlocher, D., and Zue, V., "A Model of Lexical Access from Partial Phonetic Information", *Proceedings ICASSP 1984*, #CH1945-5/84/0000-0277.
- [ITAK75] Itakura, F., "Minimum Prediction Residual Principle Applied to Speech Recognition", *IEEE Transactions Acoustical, Speech, Signal Processing*, ASSP-23, 1975, 67-72.

[KLAT74] Klatt, D.H., "Word Verification in a Speech Understanding System", in Speech Recognition: Invited Papers Of The 1974 IEEE Symposium, Reddy, D.R. ed., Academic Press, New York, 1975, 321-341.

[KRUS83] Kruskal, J. and Sankoff, D., "An Anthology of Algorithms and Concepts For Sequence Comparison", in Time Warps, String Edits, and Macromolecules: The Theory And Practice Of Sequence Comparison, Sankoff, D. and Kruskal, J. eds., Addison-Wesley, Reading, Ma., 1983, 265-311.

[LIZZ87] Lizza, Capt. G., Munger, M., Small, Capt. R., Feltshans, G., and Detro, S., "A Cockpit Natural Language Study - Data collection and Initial Data Analysis", Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, April 1987, Doc.#AFWL-TR-87-3003.

[LEVI83] Levinson, S.E., Rabiner, L.R., and Sondhi, M.M., "Speaker Independent Isolated Digit Recognition Using Hidden Markov Models", Proceedings of ICASSP Boston, 3, 1983, 1049-1052.

[LEVI87] Levinson, S.E., "Continuous Speech Recognition by Means of Acoustic/Phonetic Classification Obtained from a Hidden Markov Model", Proceedings of ICASSP Dallas, 1, 1987, 93-95.

[LOWE80] Lowerre, B. and Reddy, D.R., "The Harpy Speech Understanding System", in Trends In Speech Recognition, Lea, W.A. ed., Prentice Hall, Englewood Cliffs, N.J., 1980, 101-124.

[LYON82] Lyon, R.F., "A Computational Model of Filtering, Detection, and Compression in the Cochlea", ICASSP, May 1982, 1282-1285.

[LYON83] Lyon, R.F., "A Computational Model of Binaural Localization and Separation", ICASSP, 1983, 1148-1151.

[LYON84] Lyon, R.F., "Computational Models of Neural Auditory Processing", ICASSP, 1984, Vol. 36-1.

[MILL55] Miller, G. and Nicely, P., "An Analysis of Perceptual Confusions Among Some English Consonants", Journal of the Acoustic Society of America, 27, 1955, 338-352.

[MINE78] Mines, M.A., Hansen, B.F., and Shoup, J.E., "Frequency of occurrence of phonemes in conversational English", Lang. Speech 21, 221-241.

[MOND87] Monderer, B. and Lazar, A. A., "Detection of Speech Signals at the Output of a Cochlear Model", Proceedings 25th Allerton Conference on Communication, Control and Computing, Monticello, Illinois, 1987, 183-191.

[MOND88] Monderer, B., and Lazar, A. A., "Speech Signal Detection at the Output of a Cochlear Model", ICASSP, 63-66.

[MYER81] Myers, C.S. and Rabiner, L.R., "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition", IEEE Transactions on Acoustics, Speech, and Signal Processing, 29, 1981, 286-297.

[NEY 84] Ney, H., "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition", IEEE Transactions on Acoustics, Speech, and Signal Processing, 32, 1984, 263-271.

[OHGU88] Ohguro, Y. and Nakagawa, S., "The Relationships between Phoneme Recognition Accuracy and Sentence Recognition Accuracy", The Second Joint Meeting of ASA and ASJ, November 1988.

[OSIK75] Oskika, B.T., Zue, V.W., Weeks, R., Neu, H., and Aurbach J., "The Role of Phonological Rules in Speech Understanding Research", IEEE Trans. on Acoustics, Speech and Signal Processing, ASSP-23, No. 1, 1975, 104-112.

[PARS87] Parsons T. W., "Voice and Speech Processing", MacGraw Hill 1987, 119-125.

[PICO86] Picone, J., Goudie-Marshall, K., Doddington, G., and Fisher, W., "Automatic Text Alignment for Speech System Evaluation", IEEE Transactions on Acoustics, Speech, and Signal Processing, 34, 1986, 780-784.

[RABI86] Rabiner, L.R. and Juang, B.H., "An Introduction to Hidden Markov Models", IEEE ASSP Magazine, January 1986, 4-16.

[RUDN87] Rudnick, A., Baumeister, L., DeGraaf, K., and Lehman, E., "The Lexical Access Component of the CMU Continuous System", Proceedings ICASSP Dallas, 1, 1987, 376-379.

[SCHA81] Schank, Roger C and Reisbeck, Christopher K. *Inside Computer Understanding*, Lawrence Erlbaum Associates, Inc., 365 Broadway, Hillsdale, New Jersey, 1981, p 10.

[SAKO78] Sakoe, H. and Chiba, S., "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Transactions on Acoustics, Speech, and Signal Processing, 26, 1978, 43-49.

[SAKO79] Sakoe, H., "Two-Level DP-Matching — A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition", IEEE Transactions on Acoustics, Speech, and Signal Processing, 27, 1979, 588-595.

[SEAR79] Searle, C.L., Jacobson, J.Z., and Rayment, S.G., "Stop consonant Discrimination Based on Human Audition", J. Acoustical Society of America, Vol 65 No 3, March 1979, 799-809.

[SENE87] Seneff, S., "Vowel Recognition Based on 'Line Formants' Derived From an Auditory-Based Spectral Representation", Proceedings Speech Recognition Workshop, San Diego, CA, March 1987.

[SHEP80] Shepard, R., "Multidimensional Scaling, Tree-Fitting, and Clustering", Science, 210, October 1980, 390-398.

[SHIP82] Shipman, D.W. and Zue, V.W., "Properties of Large Lexicons: Implications for Advanced Isolated Word Recognition Systems", ICASSP 1982 Proceedings, 1982, 546-549.

[SMIT80] Smith, A.R. and Sambur, M.R., "Hypothesizing and Verifying Words for Speech Recognition", in Trends In Speech Recognition, Lea, W.A. ed., Prentice Hall, Englewood Cliffs, N.J., 1980, 101-124.

[WAIB81] Waibel, A. and Yegnanarayana, B., "Comparative Study of Nonlinear Time Warping Techniques in Isolated Word Speech Recognition Systems", Research Paper - Carnegie-Mellon University, CMU-CS-81-125, June 1981.

[WATA86] Watari, M., "New DP Matching Algorithms for Connected Word Recognition", Proceedings of ICASSP Tokyo, 2, 1986, 1113-1116.

[WINO83] Winograd, Terry, *Language as a Cognitive Process*, Addison-Wesley, Reading Massachusetts, 1983, p. 24.

[WOLF77] Wolf, J.J. and Woods, W.A., "The HWIM Speech Understanding System", IEEE International Conf. Record on Acoustics, Speech, and Signal Processing, May 1977, 784-787.

[WOOD75] Woods, W.A., "Motivation and Overview of SPEECHLIS: An Experimental Prototype for Speech Understanding Research", IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-23, 1975, 2-10.

[YODE86] Yoder, S.K., and Jamieson, L.H., "Accurate Recognition of Stop Consonants", Purdue Univ. Tech. Report, TR-EE 86-42, 1986.

[YODE87] Yoder, S.K., and Jamieson, L.H., "Speaker-Independent Recognition of Stop Consonants", ICASSP Proceedings, 1987.



MISSION of *Rome Air Development Center*

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C³I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C³I systems. The areas of technical competence include communications, command and control, battle management information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic reliability/maintainability and compatibility.